National Soil Information System 5.2
Technical Data Model Diagrams

United States Department Of Agriculture
Natural Resources Conservation Service
Information Technology Center
2150 Centre Ave. Bldg. A
Ft. Collins, CO 80526
(970) 295-5464

NASIS Project

Project Manager: Ken Harward
Soil Survey Business Area Project Manager: Terry Aho
Technical Project Manager: Gary Spivak

Introduction

The purpose of this document is to present the structure of a NASIS 5.2 database in a graphical form. The relationships between entities or tables in a database are more easily conveyed graphically than in a narrative fashion.

Prior to version 4.0 of NASIS, we produced a single database structure diagram that displayed the entire NASIS database. It fit on one legal size piece of paper, and there wasn't much room to spare. The symbology that was used did not correspond to any official data modeling diagramming standard.

With the addition of the site/point entities to NASIS 4.1, it was no longer possible to fit a complete NASIS database structure diagram on a single piece of paper that could be printed rather than plotted. Therefore, since NASIS 4.1, we have broken up the NASIS database structure diagram into a series of sub-diagrams. We now provide a high level conceptual entity relationship diagram, a high level physical data model diagram and a separate physical data model diagram for each owned object (a group of tables managed as a unit) in NASIS.

In addition, these diagrams are now rendered using one of the standard data modeling diagram symbol sets. The standard used for these diagrams is referred to as Bachman notation. This notation provides more information about relationships between entities than our previous format could.

Most data model diagrams display the names of the attributes that make up an entity, right on the diagram. We made a conscious decision not to do so because we feel that it gets in the way of focusing on the relationships between entities. The attributes that make up any of the owned objects on these physical data model diagrams can be found in the Table Structure report, which should also be available from the same location where this document is posted.

The last section of this document contains a primer on the Bachman notation used in these diagrams, as well as some common examples using this symbology that will hopefully aid in the understanding of this type of data model diagram.

Clutter Reduction Diagramming Convention

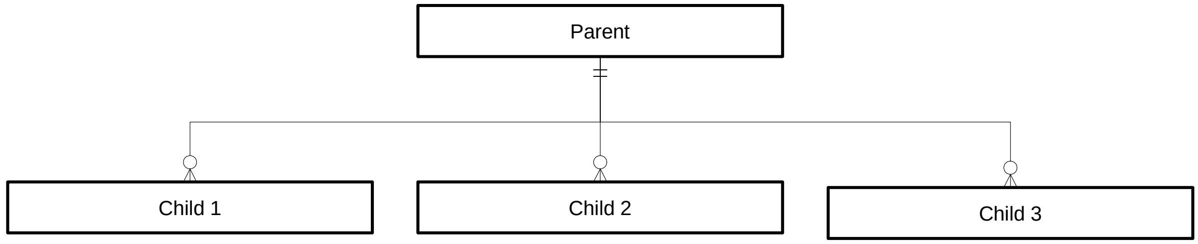
The last section of this document provides a brief introduction to the Bachman data model symbology. Throughout this document, we use an additional convention that is not a part of the Bachman standard. In order to reduce the clutter of multiple relationship lines when a common parent table has the same relationship with a number of child tables, we have the relationship lines to the common children partially coincide. Ideally in a data model, relationship lines might cross one another, but they should never share a coincidence.

In a group of tables representing one common parent table and many child tables that have the same type of relationship with the common parent, the parent table is always the table with the "one and only one" (the double cross) connector attached. The child tables all share exactly the same type of connector, usually a "zero to many" (circle and crow's foot) connector. This is not any sort of standard. This just happens to be true for every model in this document.

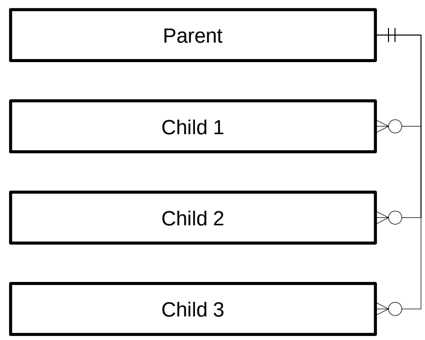
This convention is somewhat misleading because it makes it appear that there are relationships between the child tables. There may indeed be relationships between some of the child tables, but in this document the connection between two child tables that are connected by a coincidental relationship line from a common parent does not represent a relationship between those two child tables. If two children of a common parent are related, there will be a separate explicit relationship line between those two tables that has no connection to their common parent.

Please see the examples on the following page. Keep in mind that the number of child tables sharing a coincidental relationship line to a common parent may vary from 2 to N.

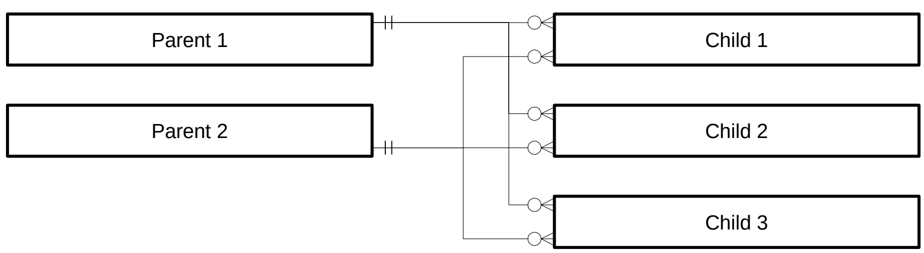
Examples of the Clutter Reduction Diagramming Convention Where Multiple Child Tables Have the Same Relationship With a Common Parent Table



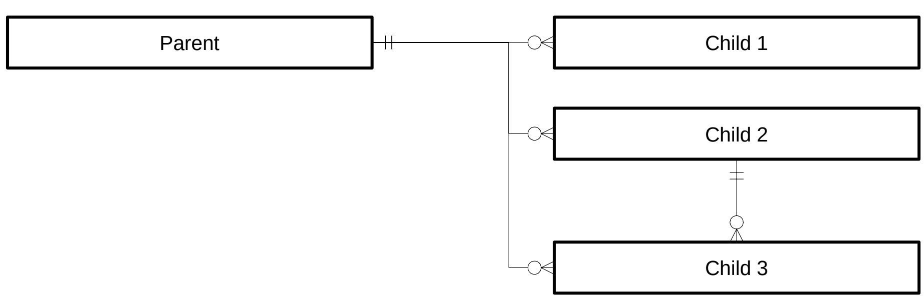
Example 1 - None of the child tables are related to one another.



Example 2 - None of the child tables are related to one another.



Example 3 - None of the child table are related to one another.



Example 4 - Child 2 and child 3 are related.

High Level Conceptual Entity Relationship Diagram

A conceptual model typically shows relationships between entities that are meaningful to a business area specialist. A conceptual model should be independent of the DBMS in which an information system is ultimately implemented. The entities in a conceptual model often resolve into more than one table when they are ultimately implemented in a DBMS.

NASIS will eventually manage four different broad categories of soil related data:

Aggregate Data (soil survey areas, map units, map unit components)

Site/Point Data (pedon descriptions, chemical and physical analytical data)

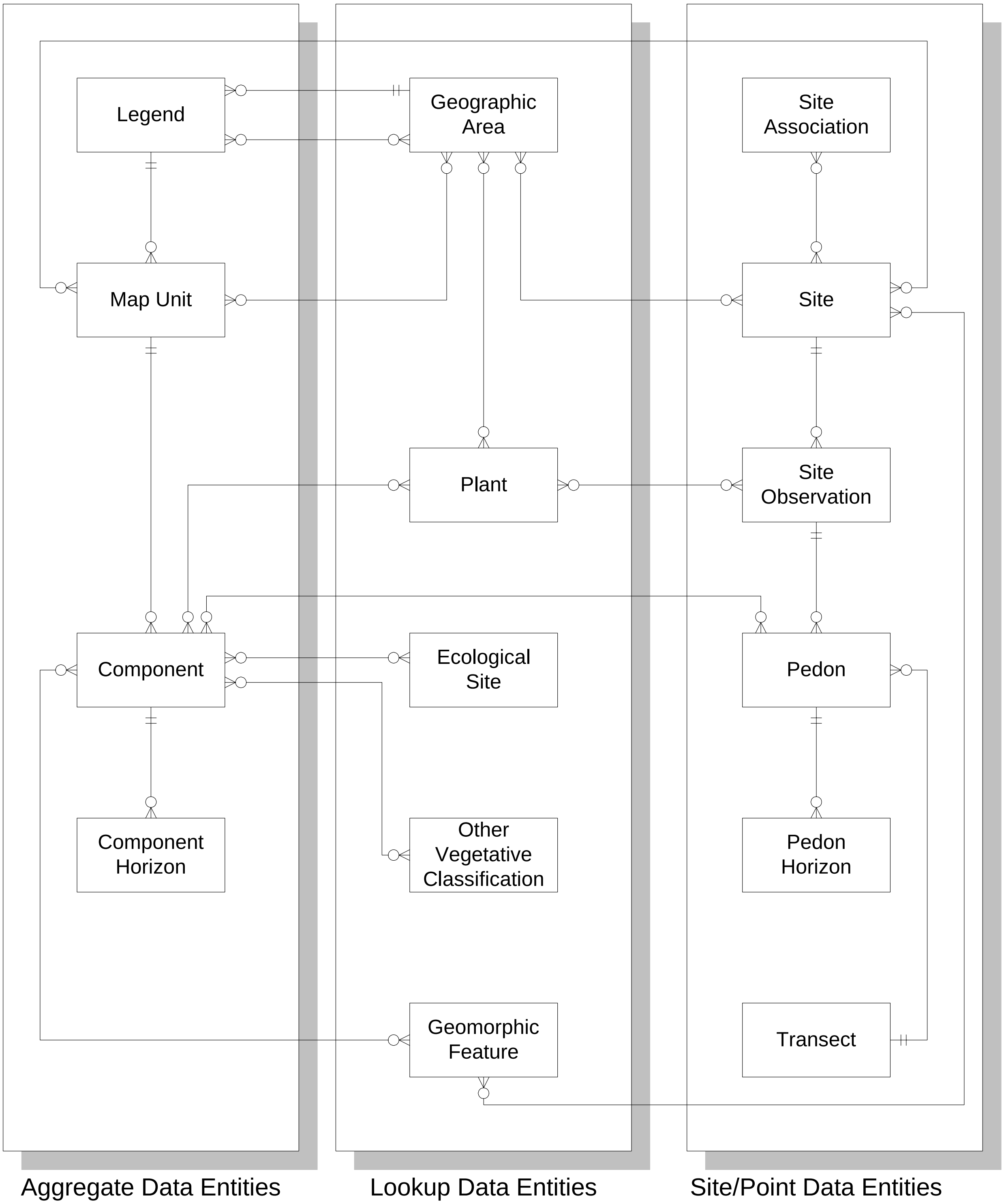
Conceptual Standards (soil taxonomy and soil series)

GIS Spatial Data (soil survey area maps and other supporting geographic area maps)

NASIS deals only with aggregate data and site/point data. Geographic areas are currently handled in a tabular (non-GIS) manner.

The entities in this diagram are grouped into three categories, entities that represent aggregate data, entities that represent site/point data, and entities that serve as lookup tables (domains). This last category includes entities that are not exclusive to NASIS. The geographic area domain is common to almost all NRCS applications. The plant and ecological site domains are shared among multiple NRCS applications.

NASIS 5.2 High Level Conceptual Entity-Relationship Diagram



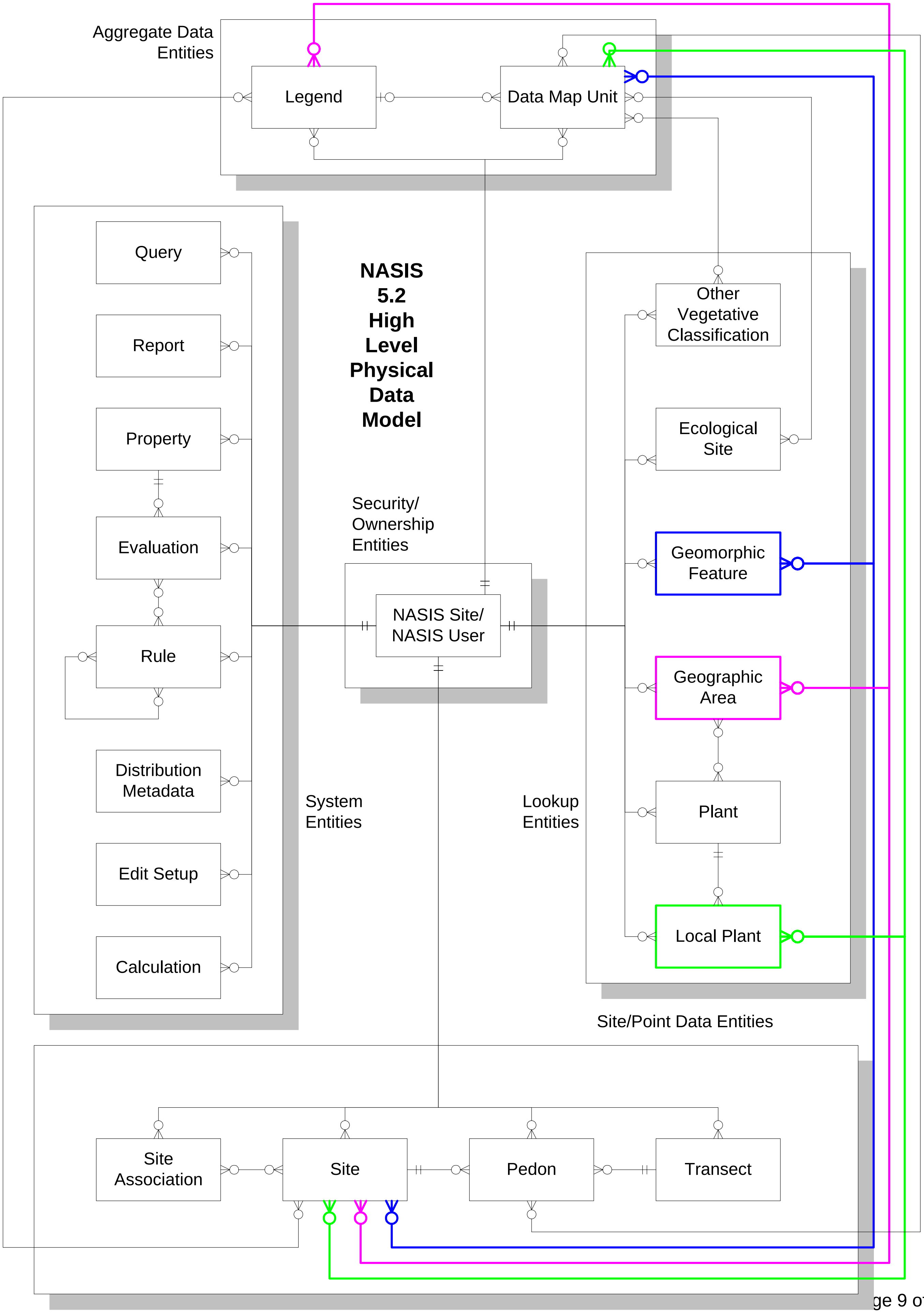
High Level Physical Data Model

Most people would argue that the term “High Level Physical Data Model” is an oxymoron, but we of the NASIS staff have never been afraid to push the inside of the envelope. In NASIS, tables are grouped together into things we refer to as “owned objects”. An owned object must contain one or more tables, but a table must be a member of one, and only one, owned object.

NASIS manages security and access at the owned object level. In other words, if a person has write access for a particular owned object, they have write access to any table that is a member of that owned object. When a person loads an instance of an owned object into the NASIS editor, all of the records in all of the tables for that instance of that owned object are immediately locked. Another NASIS user may read that same instance of that locked owned object, but no user can modify that instance of that owned object until the person who has it locked releases their lock. This scheme is referred to a pessimistic locking.

Each entity in the NASIS High Level Physical Data Model corresponds to one (or in one case, two) of the 22 owned objects in NASIS. The entities in this diagram are grouped into five categories, aggregate data entities, site/point data entities, lookup entities, system entities and security/ownership entities. A detailed description of the tables that make up each of these owned objects is found in the Physical Data Model Diagrams section.

In NASIS 5.0 the NASIS Site owned object was split into NASIS Site and NASIS User. Since NASIS 5.0, NASIS users are independent of NASIS sites. The entity “NASIS Site/NASIS User” represents both of these entities. It just so happens that the NASIS Site entity has the same relationship to all non-security NASIS owned objects as does NASIS User, almost. Each owned object has a mandatory relationship to NASIS Site but an optional relationship to NASIS User. These two entities were combined in this diagram because showing both entities independently, as well as all of their relationships, would have cluttered up the diagram beyond readability. The purpose of this diagram is to provide a “relatively” uncluttered view of the relationships that exist between the 22 NASIS owned objects.


Aggregate Data Entities
Legend
Data Map Unit
Query
Report
Property
Evaluation
Rule
Distribution Metadata
Edit Setup
Calculation
NASIS 5.2 High Level Physical Data Model
Security/ Ownership Entities
NASIS Site/ NASIS User
System Entities
Lookup Entities
Other Vegetative Classification
Ecological Site
Geomorphic Feature
Geographic Area
Plant
Local Plant
Site/Point Data Entities
Site Association
Site
Pedon
Transect

Physical Data Model Diagrams

A physical data model diagram displays physical tables and relationships that exist in a physical NASIS database. Not all of the tables nor all of the relationships that exist in a NASIS database are shown. The reason for this is discussed the NASIS Database Structure Guide document.

All of the tables that are visible in the NASIS editor are included on these diagrams, as well as a few which are not visible. All of the edit and traversable relationships in a NASIS database are included on these diagrams, as well as a few non-traversable relationships. An edit relationship exists between two tables in the same owned object where a user can navigate between these two tables using the “up table” and “down table” arrows in the NASIS editor. A traversable relationship is a non-edit relationship between two tables, not necessarily in the same owned object, that can be navigated via the “Load Related” and “Find Related” menu items in the NASIS editor. A non-traversable relationship, while not navigable in the NASIS editor, can be referenced and used in queries and reports. Referential integrity for all relationships in NASIS is enforced at the database engine level, i.e. all relationships are defined as permanent referential constraints.

This page unintentionally left blank.

Geographic Area Owned Object

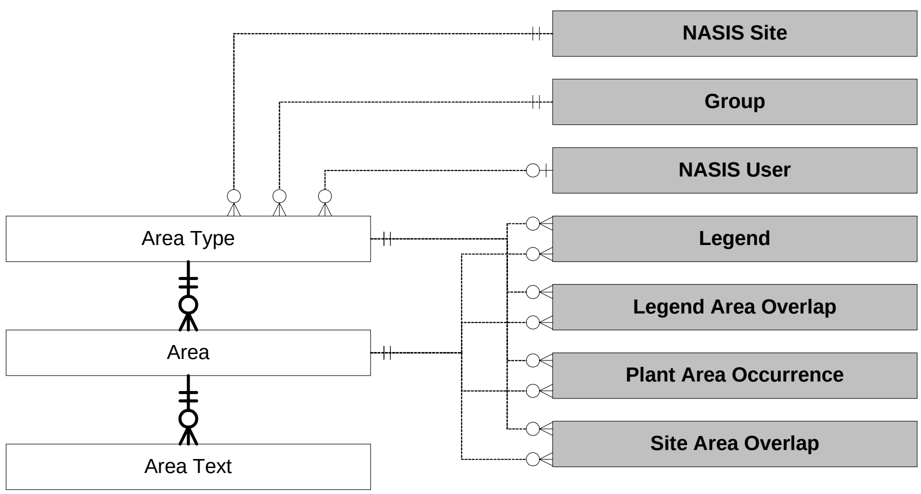
Because we intend to eventually integrate the tabular NASIS database with a GIS, we have taken great care to be very clear about what entities will ultimately correspond to some GIS polygon, line or point. In other words, rather than recording a relationship with a spatial entity as a tabular attribute of some entity, we create an independent entity to represent that spatial entity and then establish a relationship between it and the other related entity.

There are only three tables in NASIS that represent a spatial entity. These three tables are the Area table, the Mapunit table and the Site table. With the exception of the Mapunit table, all other spatial entities that correspond to one or more polygons are recorded in the Area table. The Mapunit table has special status because it is a fundamental part of a soil survey that has many more attributes than other more generic areas. For generic areas recorded in the Area table, only an area symbol, an area name and the extent of that area in acres are recorded. In the generic Area table, an area symbol must be unique within the context of it parent Area Type record.

The geographic area model in NASIS is a very flat model with limited capabilities. This was a conscious decision to not try to emulate very many GIS capabilities in a tabular database, since that tabular database will ultimately be integrated with a GIS. NASIS cannot record hierarchies of areas. NASIS cannot record the coincidence of one generic area with another generic area.

The Site table is the only table in NASIS that represents a point. Unlike the case for polygons, there is no generic table in NASIS to record other points of interest.

NASIS 5.2 Geographic Area Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Legend Owned Object

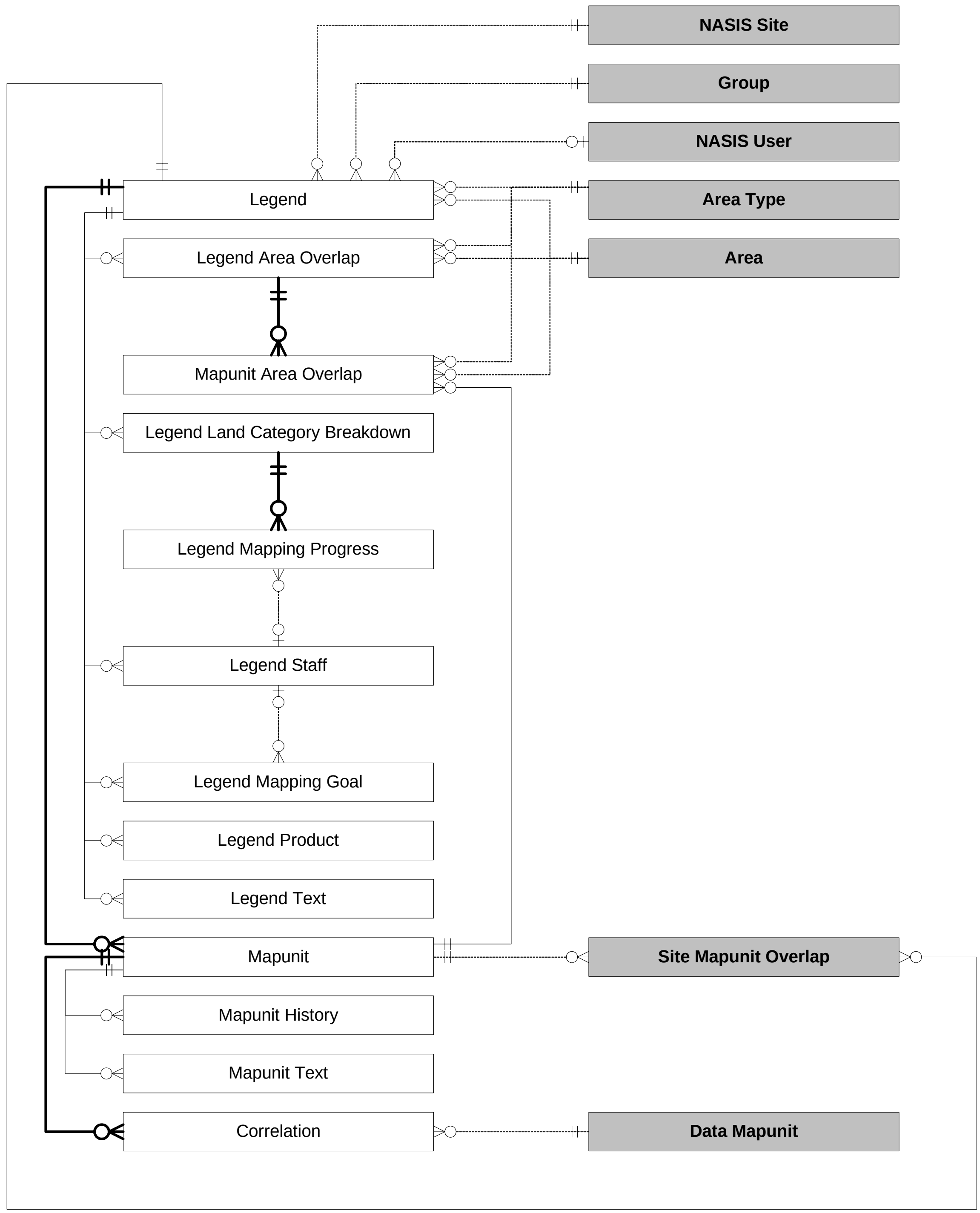
The Legend owned object represents an instance of a soil map that typically contains one or more map units. A legend may represent a traditional soil survey area, an MLRA soil survey, or any other ad hoc soil survey area. A legend must identify one and only one area in the Area table. While NASIS area types include Non-MLRA Soil Survey Areas (traditional soil survey areas) and MLRA soil survey areas, a legend may correspond to any area for any area type that exists in the Area Type table.

NASIS has the capability to record more than one vintage of a legend for the same soil survey area. This means that a NASIS database can contain both an out of date legend and the current legend for the same soil survey area.

Any number of coincidences between a legend and some other geographic area may be recorded. The Legend owned object is the only place in NASIS where a coincidence of three different geographic areas may be recorded. A record in the Mapunit Area Overlap table represents the coincidence of a soil survey area, a map unit within that soil survey area and some other geographic area.

The Legend owned object records the map units that are a part of that legend, but not the actual soil attributes associated with those map units. There is a relationship between a map unit and the entity that contains the soil attribute data associated with that map unit. The owned object that contains the soil attribute data associated with a map unit is the Data Mapunit owned object. The reason NASIS was designed this way was to solve the problem associated with the artificial boundaries of soil survey areas. By separating the concept of a map unit from the soil attribute data that represents that map unit, we can associate the same Data Mapunit with map units in two adjacent soil surveys. This permits those two map units that are represented by the same attribute data to have different map unit symbols, and provides the capability to make a perfect join across soil survey area boundaries without having to duplicate data.

NASIS 5.2 Legend Owned Object



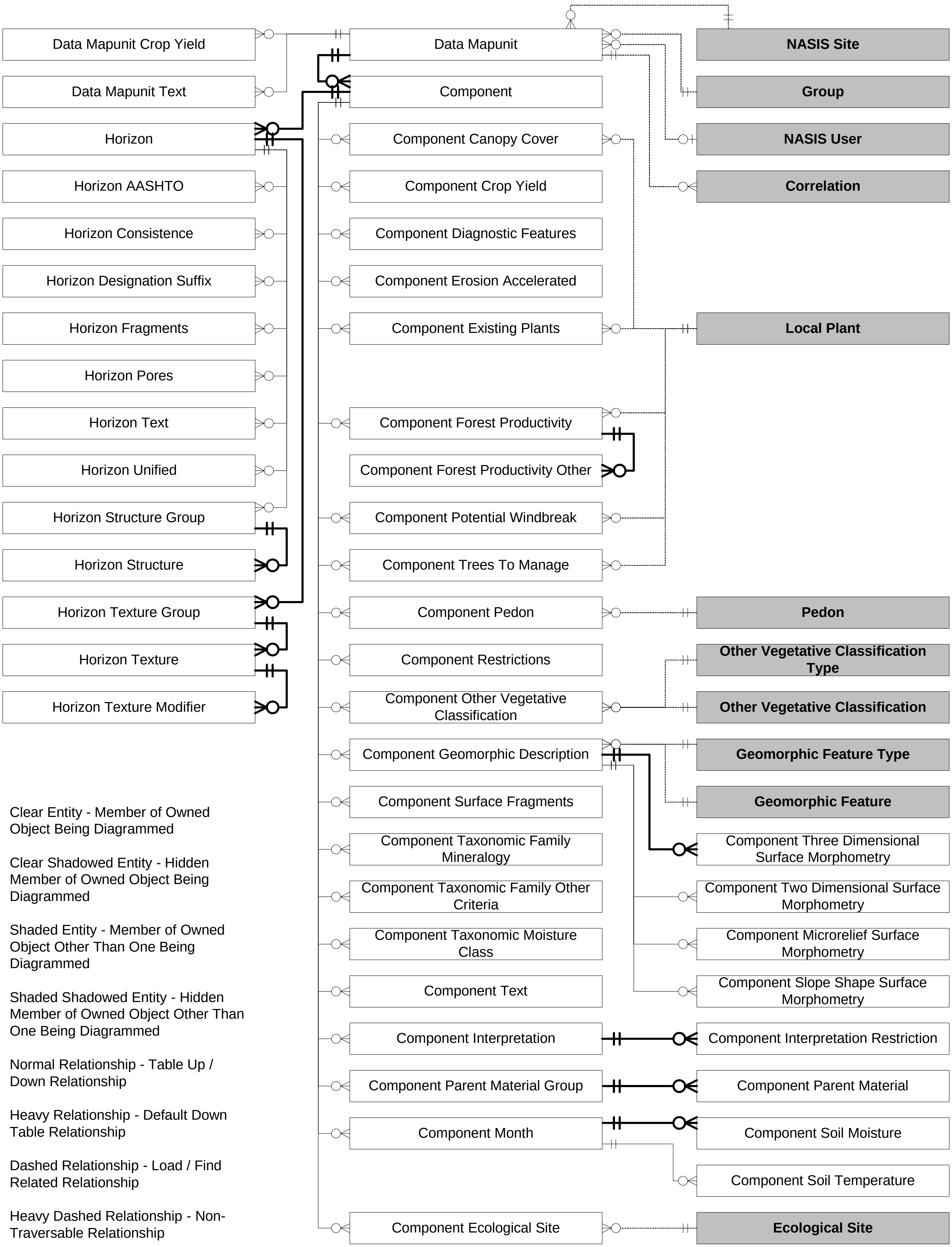
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Data Mapunit Owned Object

A Data Mapunit owned object represents the typical range in characteristics of a map unit that occurs in one or more soil survey areas. In theory, a data mapunit in NASIS is not georeferenced, per se. A data map becomes georeferenced when it is related to a particular map unit. In practice, a data mapunit is constructed with some general geographic area in mind. In reality, this distinction is fuzzy, at best.

The separation between a map unit and a data mapunit permits the same soil attribute data to be associated with two different map units, with possibly two different symbols, facilitating a perfect join across a soil survey area boundary, without have to duplicate soil attribute data. In creating such a join, a NASIS user must be careful when editing a data mapunit to make sure that those edits apply to all locations where that data mapunit is referenced. The intent of the NASIS designers was that multiple references to the same data mapunit would only occur along the boundaries of adjacent soil survey areas. There is nothing in NASIS that prevents a data mapunit that is referenced by a soil survey area in Alaska from also being referenced by a soil survey area in Florida.

NASIS 5.2 Data Mapunit Owned Object



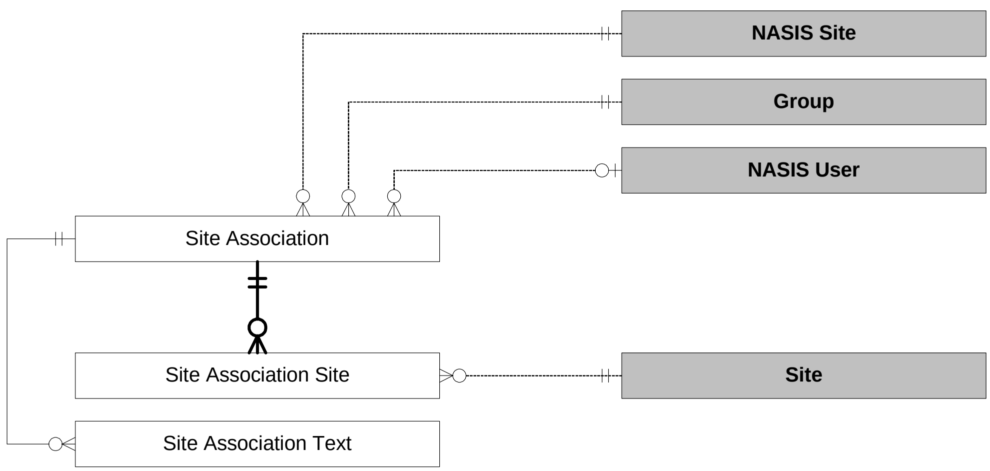
Site Association Owned Object

The Site Association owned object provides a way to associate a group of sites (Site owned object). A site association may contain one or more sites, and a site may be a member of one or more site associations. A site is not required to be a member of any site association.

One might want to associate a group of sites for several reasons. The original reason that the Site Association owned object was created was to provide a way to group the sites that make up a transect. In NASIS 5.0, the concept of a transect was separated out into its own owned object, but the site association object was also retained.

As of NASIS 5.0, the primary purpose of a site association is to group sites that are in close proximity, or group sites that share some common characteristic, such as all sites that occur in a particular soil in a soil survey area, or are part of the same study.

NASIS 5.2 Site Association Owned Object

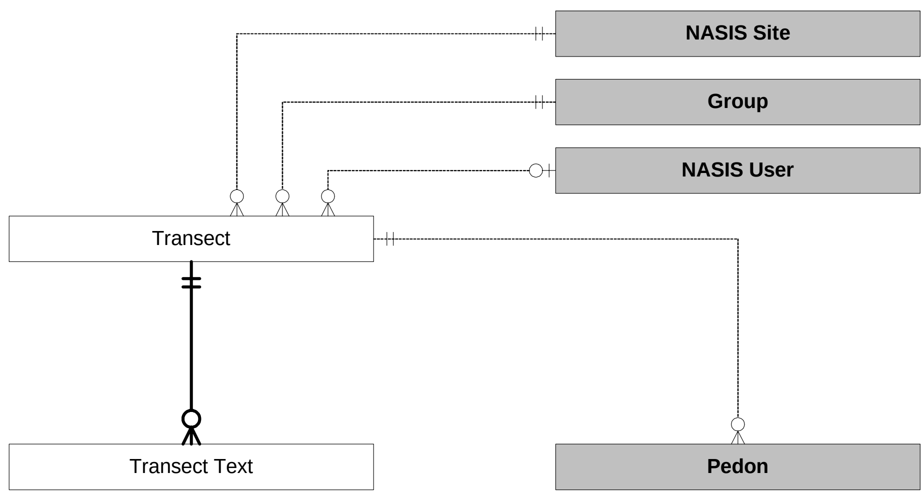


Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Transect Owned Object

The Transect owned object was added in NASIS 5.0. Prior to NASIS 5.0, the Site Association object was used to record a transect. The main problem with recording transects prior to NASIS 5.0 was that it wasn't possible to know which pedon corresponded to which transect stop number, since prior to NASIS 5.0, transect stop number resided in the Site table. Also, if more than one pedon was recorded for a site that was related to a transect, there was no way of knowing which of those pedons participated in the transect. In NASIS 5.0, transect stop number was moved to the Pedon table, and a direct relationship between a transect and its associated pedons was established. A transect may be associated with multiple pedons (of course), but a pedon can be a member of one and only one transect.

NASIS 5.2 Transect Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

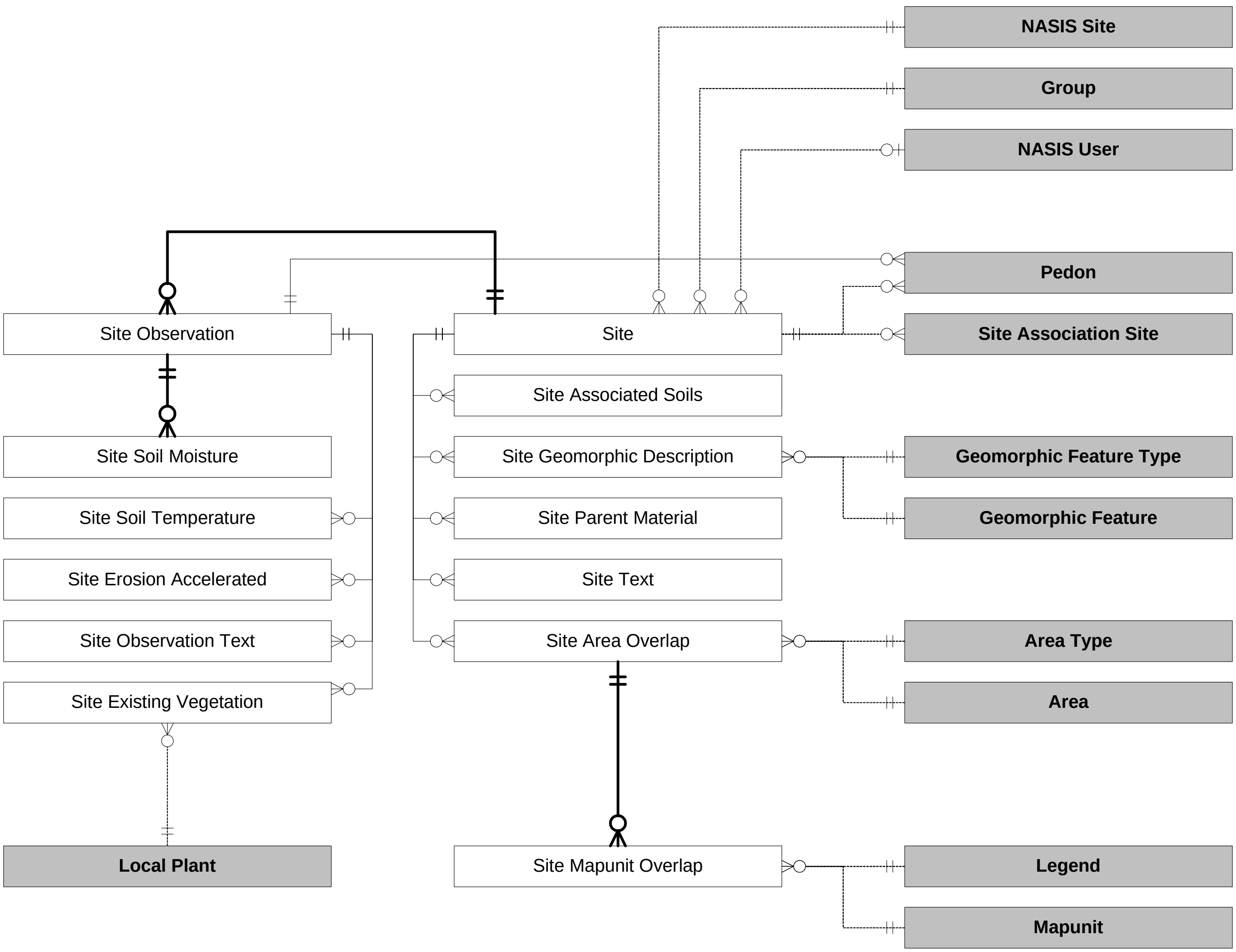
Site Owned Object

The Site owned object encompasses two concepts, a point on the ground somewhere on planet Earth, and the concept of making an observation at that point at a particular time. Traditionally in the Pedon data entry program, the concept of a transect, a site and a pedon were all bundled together. At a particular site, we may make several different types of natural resource inventories. In addition to describing a pedon, we may make either a rangeland plant inventory (Range 417) or a woodland inventory (Wood 5). Up to now, each type of inventory used a different program, and all of them had to record duplicate information about the site.

In NASIS 4.1, we separated the concept of a site and the types of natural resource inventories that might be observed at that site on a particular date. In order to do this, we had to determine what attributes would be recorded for a site, regardless of what type of natural resource inventory was being recorded. In addition, we had to make a judgement call about what attributes of a site would be considered constant, and what attributes of a site could change over time. Those attributes of a site that we decided to treat as non-varying can obviously vary over a long enough period of time.

While we define a site as a point on the ground, obviously some areal extent is associated with a site. In the case of a pedon, it is roughly from one to ten square meters. In the case of a Wood 5, it may be a tenth acre. Nevertheless, we choose to treat it as a point, and we record a single longitude and latitude for a site. Typically, we would not expect more than one pedon to be associated with a site on the same date, but the data model does not prevent this. If a user wants to describe a pedon with several close satellite pedons and treat them as the same site, they are free to do so.

NASIS 5.2 Site Owned Object



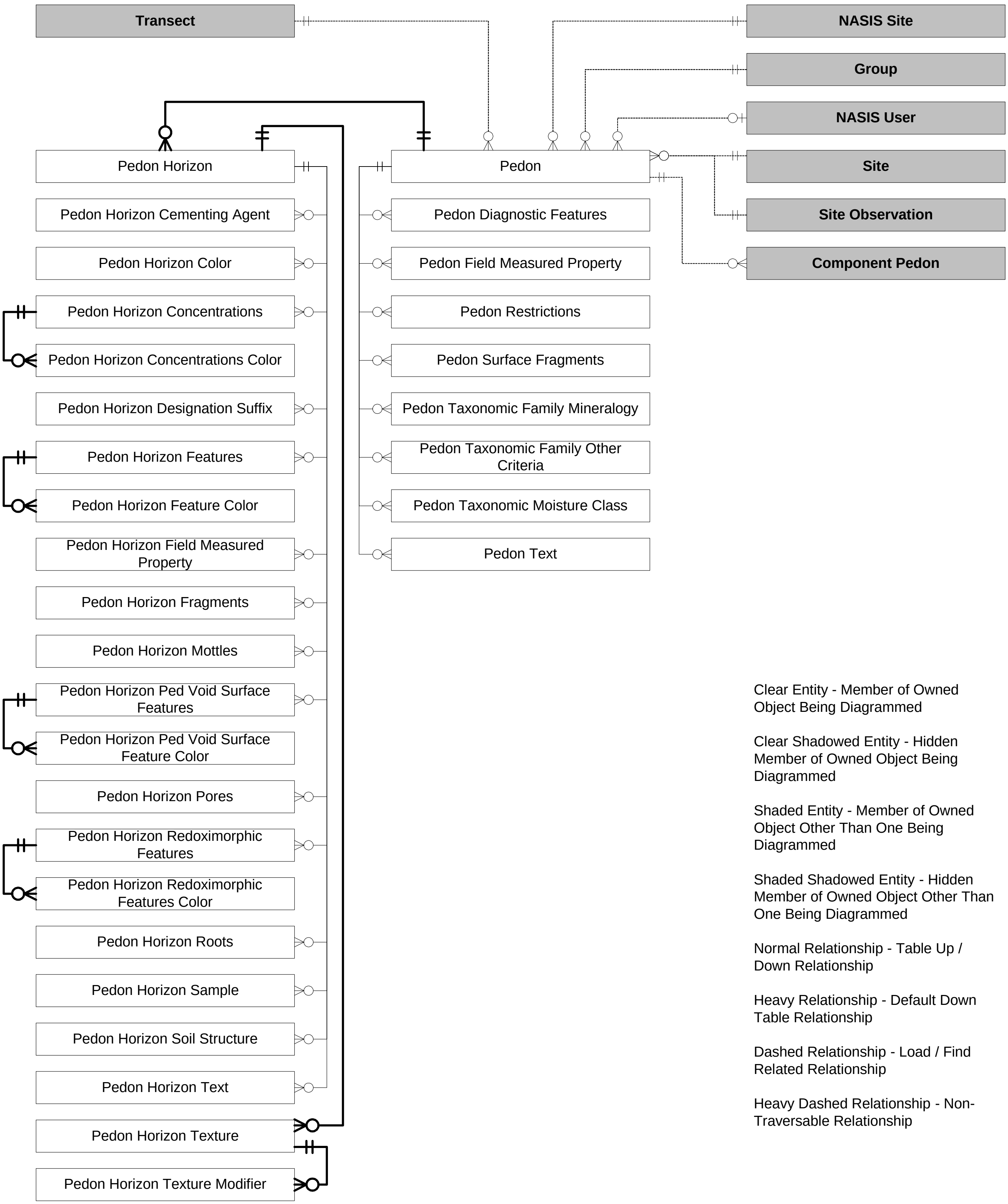
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Pedon Owned Object

A Pedon owned object records the morphological description of a particular pedon, at a particular site, at a particular time. There are many data structure similarities between pedons and data mapunits. A pedon represents an actual observation, whereas a data mapunit represents an aggregation of pedons that express a range of variability for a particular soil. Thus a pedon typically records a single value for an attribute, whereas a data mapunit may record a high, low and representative value for the same attribute.

A map unit component may be associated with one or more pedons, and, less frequently, a pedon may be associated with more than one map unit component. The typical relationship between a map unit component and a pedon represents a geographic coincidence, i.e. the pedon that was described resided in the map unit containing the related map unit component. The other relationship that may exist between a map unit component and a pedon is one where the pedon participated in the establishment of the soil series represented by the map unit component. In this case, there may or may not be a geographic coincidence between the pedon and the map unit component.

NASIS 5.2 Pedon Owned Object



Clear Entity - Member of Owned Object Being Diagrammed

Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed

Shaded Entity - Member of Owned Object Other Than One Being Diagrammed

Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed

Normal Relationship - Table Up / Down Relationship

Heavy Relationship - Default Down Table Relationship

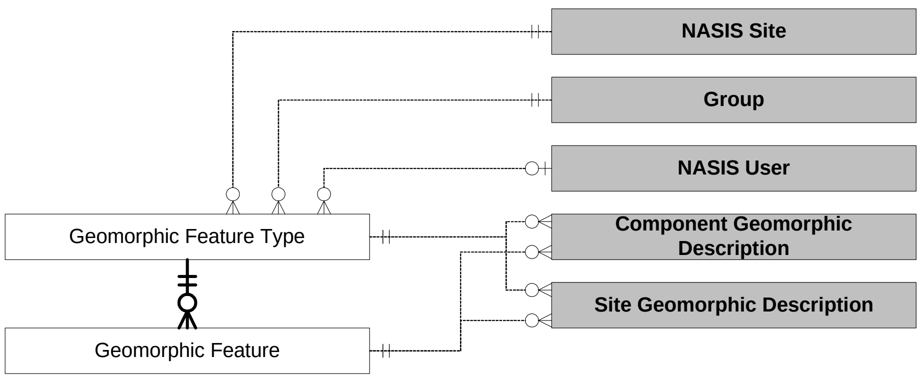
Dashed Relationship - Load / Find Related Relationship

Heavy Dashed Relationship - Non-Traversable Relationship

Geomorphic Feature Owned Object

The Geomorphic Feature owned object serves as a lookup table for geomorphic feature terms. Geomorphic features are described for both map unit components and sites. There are four categories of geomorphic feature terms, landscapes, landforms, anthropogenic (man made) features and micro-features. The NASIS site “pangaea” is the only site that can create or edit geomorphic feature terms. The geomorphology staff at the National Soil Survey Center in Lincoln NE has the responsibility for maintaining these domains.

NASIS 5.2 Geomorphic Feature Owned Object



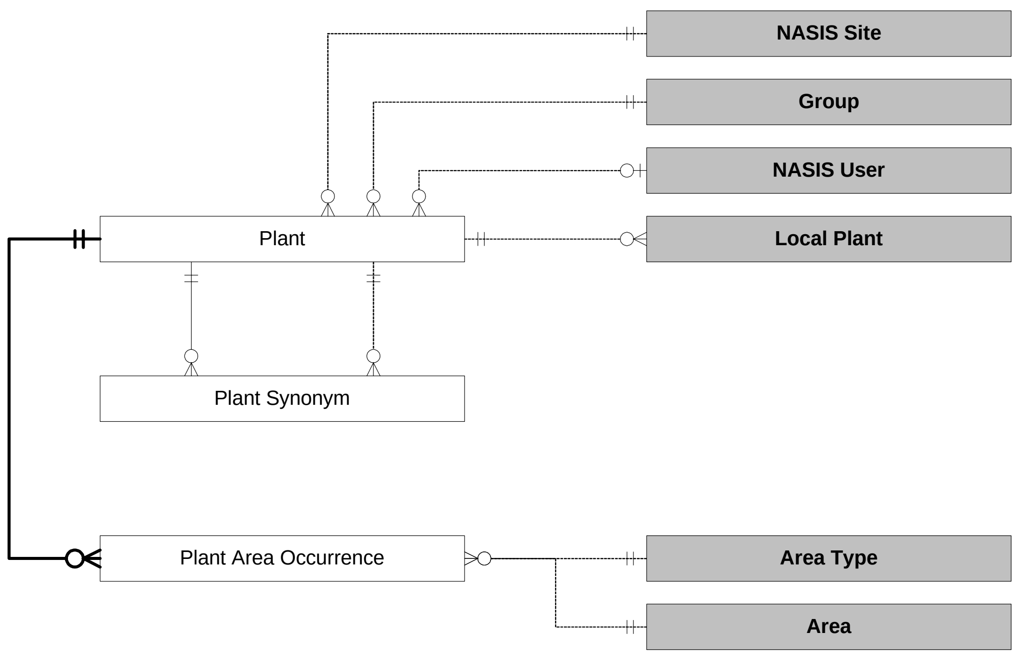
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Plant Owned Object

The Plant owned object serves as a lookup table for references to official plants. Both accepted and synonym (obsolete) plant symbols are included.

The Plant table, like the Ecological Site table, can only be edited by members of the NASIS site “flora”. The contents of the Plant table reflects the contents of the official NRCS National Plants Database, an information system external to NASIS. Similar to how the Ecological Site table is managed, this table will be periodically updated to reflect the current contents of the NRCS National Plants Database. Any change to the Plant table will merely reflect a change that has already been made to the NRCS National Plants Database.

NASIS 5.2 Plant Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Local Plant Owned Object

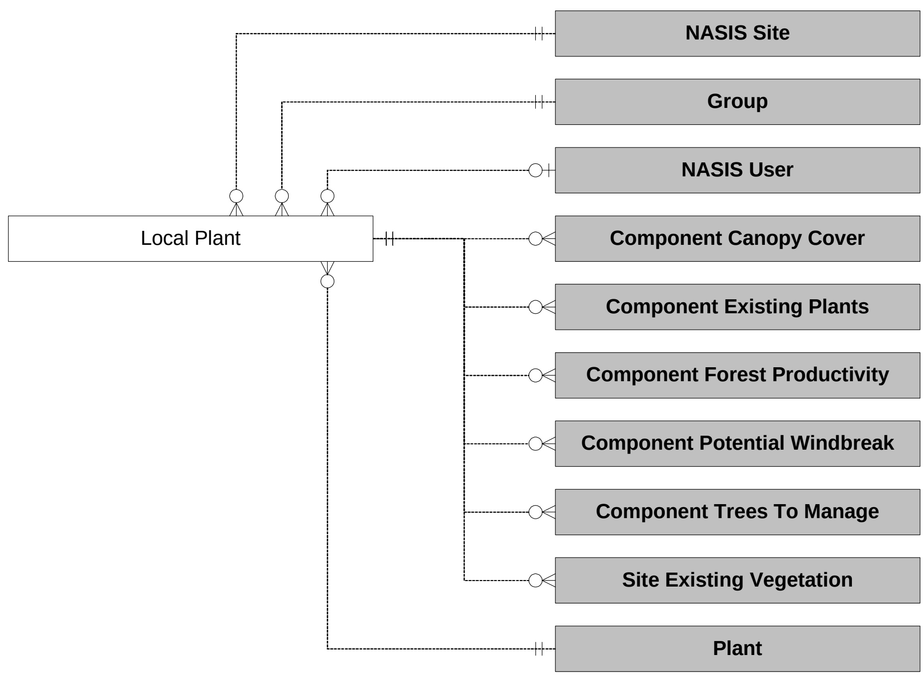
A Local Plant owned object serves as a middleman between a NASIS table that needs to reference a particular official plant, and the official Plant table itself. The Local Plant table was originally created to help facilitate the process of changing from using an unofficial plant lookup table that was managed by the national Soils development staff to using the official plant list that is managed by the National Plants development staff. The source of the Local Plant table was the old unofficial plant table that was part of the State Soil Survey Database (SSSD), the predecessor to NASIS.

The conversion from unofficial plants to official plants was made by attempting to match on plant symbol and scientific name. When a match was found, a relationship was established between the Local Plant table and the corresponding plant in the official Plant table. When no match was found, a relationship was established from the Local Plant table to a bogus record in the official Plant table that was defined as an “unknown plant”. This provided a way to start the process of moving to exclusive use of the official Plant table, without a NASIS database manager having to resolve all conflicts prior to conversion.

After this conversion, a user could select all Local Plant records that were linked to the “unknown plant” record in the official Plant table, and begin to resolve the mismatches. To aid in the resolution of these mismatches, all of the columns from the old unofficial plant table were copied into the Local Plant table.

The Local Plant table is somewhat problematic. In creating a new record in the Local Plant table, the NASIS user is responsible for the duplicate entry of the corresponding official plant symbol. On the other hand, the Local Plant table provides a way to record a common name for a plant that may be more appropriate than the common name that is available in the Plant Area Occurrence table. At this time, it is not clear whether the Local Plant table will be dropped after all mismatches are accounted for, or retained because it permits a NASIS user to enter a more appropriate regional common name for a plant.

NASIS 5.2 Local Plant Owned Object



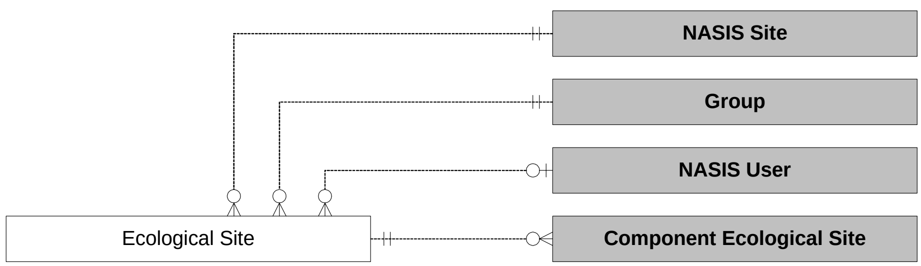
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Ecological Site Owned Object

The Ecological Site owned object serves as a lookup table for references to official NRCS ecological sites. Soil survey data may also make references to ecological sites defined by organizations other than NRCS, but these non-NRCS ecological sites are recorded using the Other Vegetative Classification owned object.

The Ecological Site table, like the Plant table, can only be edited by members of the NASIS site “flora”. The contents of the Ecological Site table reflects the contents of the official NRCS Ecological Site Description System, an information system external to NASIS. Similar to how the Plant table is managed, this table is periodically updated to reflect the current contents of the NRCS Ecological Site Description System. Any change to the Ecological Site table will merely reflect a change that has already been made to the NRCS Ecological Site Description System.

NASIS 5.2 Ecological Site Owned Object

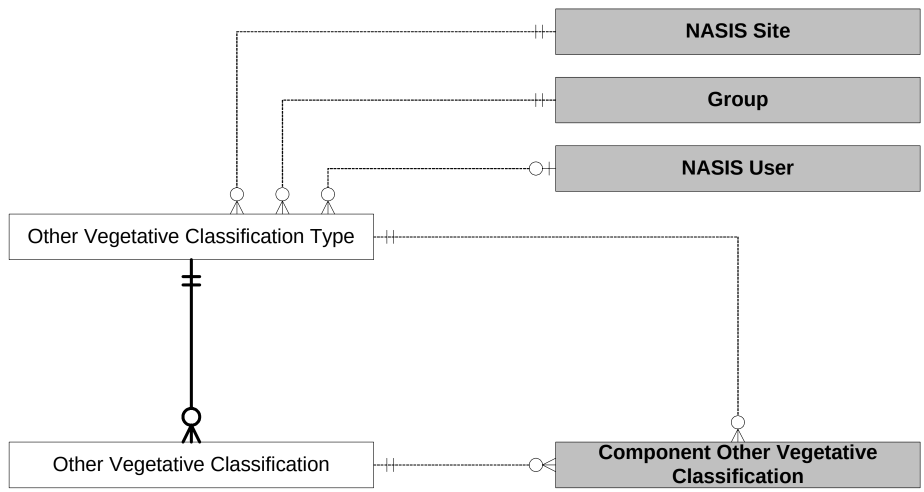


Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Other Vegetative Classification Owned Object

The Other Vegetative Classification owned object was added in NASIS 5.0. This object was added to facilitate the recording of ecological and vegetative system classifications other than ecological sites as defined by NRCS, such as U.S. Forest Service ecological site and plant community classifications. Unlike official NRCS ecological sites, any valid NASIS user with write privileges may add new other vegetative classification types and other vegetative classifications.

NASIS 5.2 Other Vegetative Classification Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

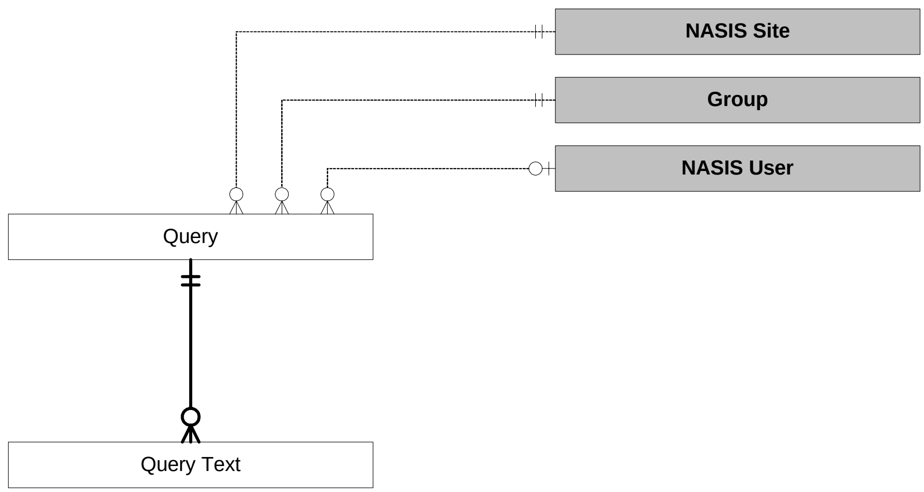
Query Owned Object

The Query owned object records user generated queries that are used for the purpose of loading data into the NASIS editor. NASIS provides a special editor for the construction of queries. There is no select clause in a NASIS query because a NASIS query always results in the loading of entire records into some target table.

A user can assign a name to a query, but that name must be unique within the context of the NASIS site in which that user is working.

The UNIX utilities Lex and Yacc were used to create a custom NASIS query language. A NASIS query is ultimately converted to a true SQL query before it is executed. The NASIS custom query language mainly provides a user with a way of making joins between tables without having to know exactly what columns participate in the join. It does this by letting the user select a predefined named join clause into the query being constructed.

NASIS 5.2 Query Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Report Owned Object

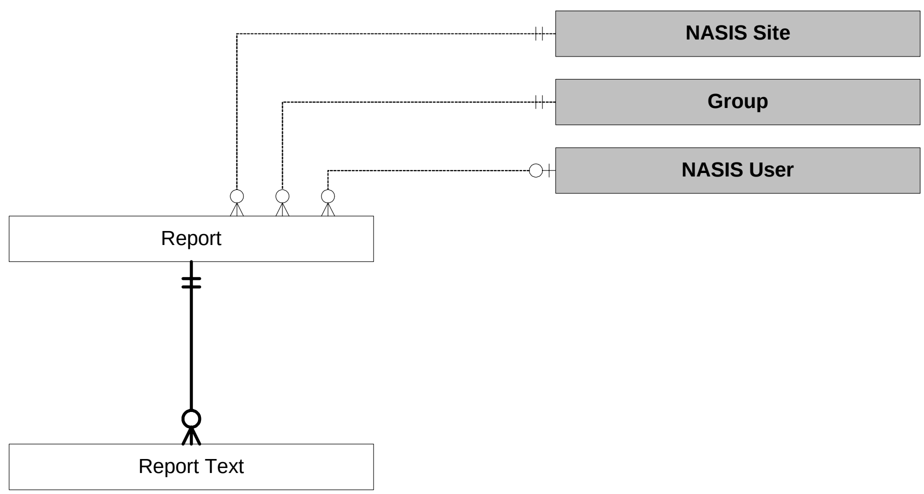
The Report owned object records a report script that generates a formatted report that may either be printed or saved to a file. NASIS does not provide a special editor for the construction of report scripts. To create a report script, a user must manually type the script into a text field. A NASIS report scripting language guide is available to anyone who is interested in creating their own custom reports.

There is a special type of report script that can be associated with a NASIS owned object. This type of report script generates a report, on demand, in the report window in the NASIS editor. At this time, only one such report can be defined for a particular owned object. A general NASIS user cannot currently create such a report. Such reports can currently be created only by the NASIS development staff.

A user can assign a name to a report, but that name must be unique within the context of the NASIS site in which that user is working.

The UNIX utilities Lex and Yacc were used to create a custom NASIS report scripting language.

NASIS 5.2 Report Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Property Owned Object

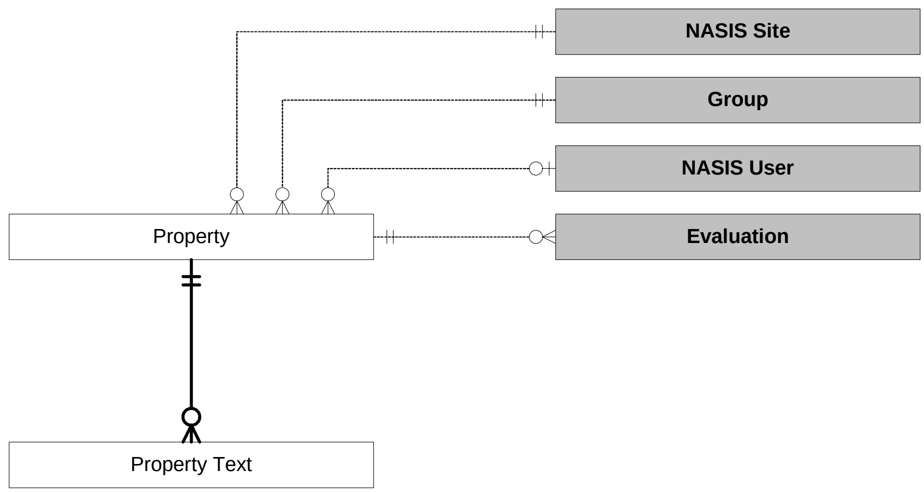
The Property owned object is a scripted procedure that is capable of returning one or more values. Computation of the return values is based on algorithms that reference permanent NASIS columns and procedure defined constants. NASIS does not provide a special editor for the construction of a property script. To create a property script, a user must manually type the script into a text field. The property scripting language is a subset of the report scripting language. Thus the report language guide can serve as a reference for the construction of properties.

Both reports (Report owned object) and evaluations (Evaluation owned object) may reference properties (Property owned object). One can think of a property script as a way to generate virtual attributes on demand. The data model diagram does not show a relationship between a property and a report because property names are embedded in the source text of a report. In other words, one cannot join the Property and Report tables. A relationship such as this is not documented on a physical data model diagram.

A user can assign a name to a property, but that name must be unique within the context of the NASIS site in which that user is working.

The UNIX utilities Lex and Yacc were used to create a custom NASIS property scripting language. This same language is also used to implement calculations and validations in NASIS. This language includes a few extensions that apply only to calculations and validations.

NASIS 5.2 Property Owned Object



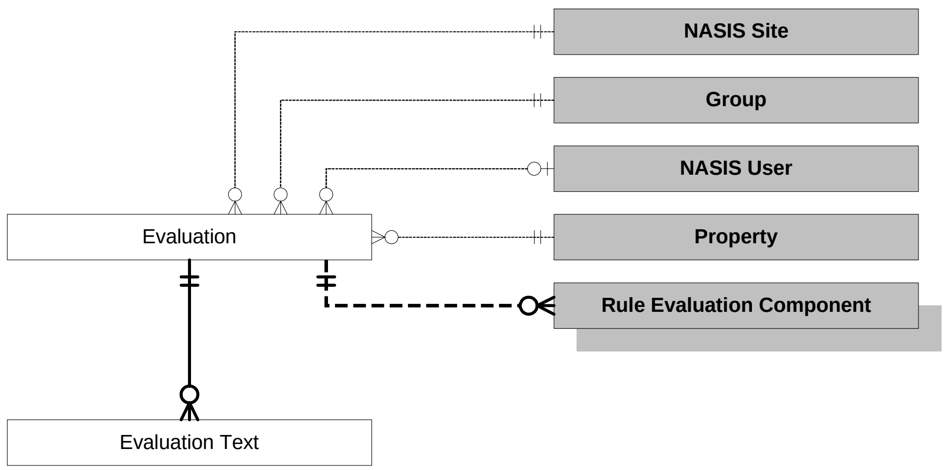
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Evaluation Owned Object

An Evaluation owned object is a transformation function that converts the value of a property (Property owned object) into the zero to one domain range utilized by the NASIS fuzzy logic interpretation generation module. One, and only one, property is used as input to the evaluation transformation function. However, more than one evaluation transformation function may exist for the same property. A variety of standard transformation functions are available, and a NASIS user may define their own transformation curve. NASIS provides a special graphical editor for the construction of evaluations.

A user can assign a name to an evaluation, but that name must be unique within the context of the NASIS site in which that user is working.

NASIS 5.2 Evaluation Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

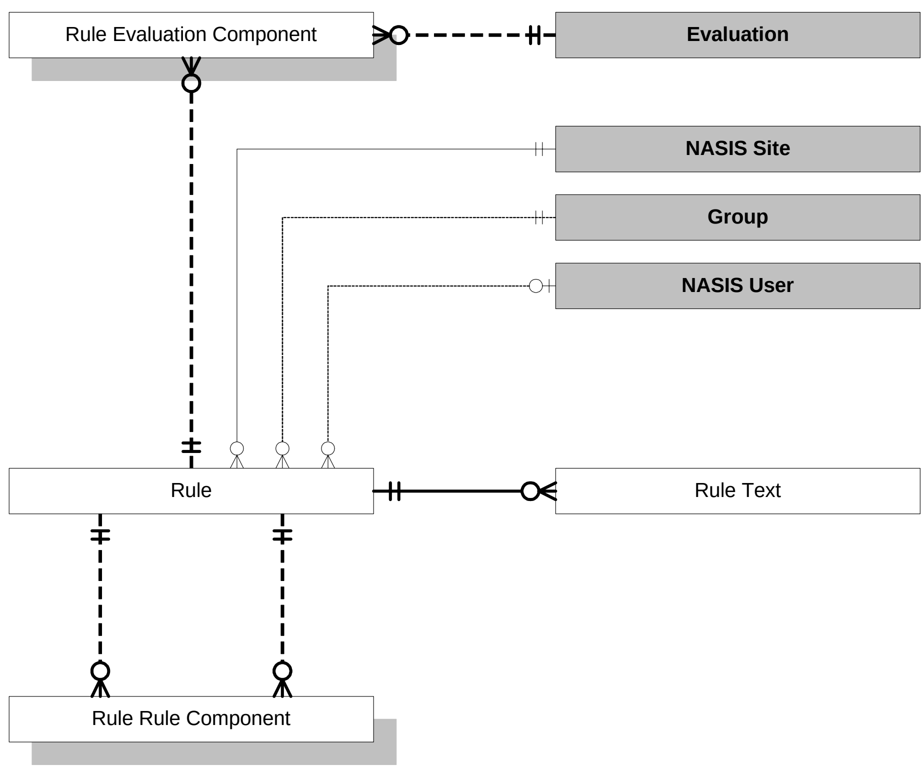
Rule Owned Object

A Rule owned object is an expression of one or more evaluations (Evaluation owned object) and sub-rules (Rule owned object) that resolves into a set of values in the zero to one domain range utilized by the NASIS fuzzy logic interpretation generation module. NASIS provide a special graphical editor for the construction of rules.

In the NASIS fuzzy logic interpretation generation module, a Rule is equivalent to an interpretation. The evaluations and sub-rules used to derive the root rule are equivalent to interpretation restrictions, i.e. why this interpretation value came out the way it did. Although the output of a Rule is a set of values in the range of zero to one, those values may be "defuzzified" into discrete classes, and a label may be assigned to each of those classes. For example, one could defuzzify a fuzzy logic value into "slight", "moderate" and "severe" classes.

Prior to NASIS 5.0, a rule name had to be unique within the context of a particular NASIS site. As of NASIS 5.0, a rule name must be unique among ALL NASIS sites.

NASIS 5.2 Rule Owned Object

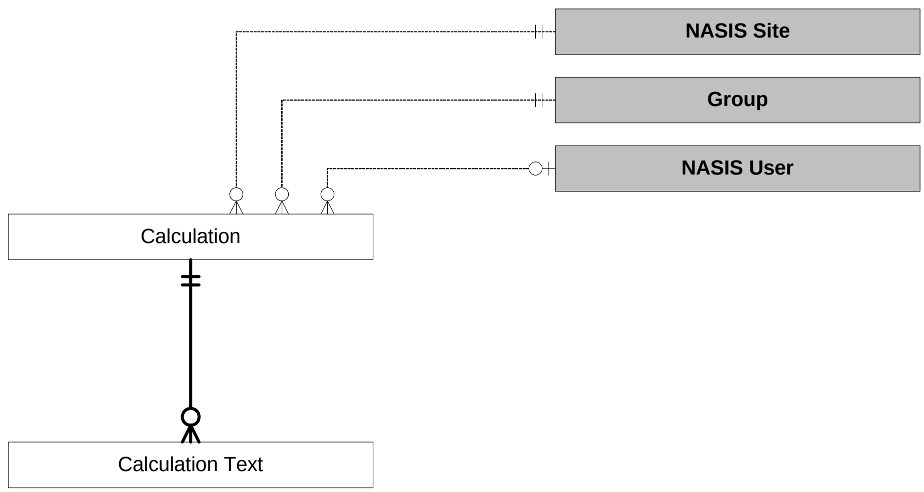


Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Calculation Owned Object

The Calculation object was established in NASIS 5.0. A calculation data dictionary table has always existed in NASIS. Since NASIS 5.0, this table is now accessible via the NASIS editor. The Calculation object records both calculations and validations. A calculation stores results back into one or more columns, whereas a validation reports discrepancies but does not update any underlying data in the database. Only members of the NASIS site “pangaea” can add or edit Calculations.

NASIS 5.2 Calculation Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

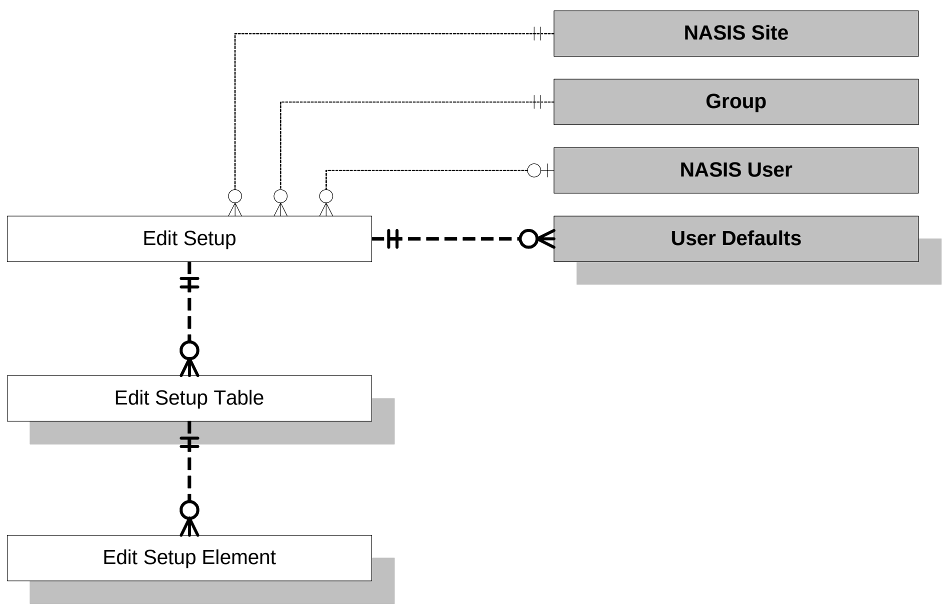
Edit Setup Owned Object

The Edit Setup owned object provides a NASIS user a way to customize how data is presented in the NASIS editor. Before NASIS 4.1, the user had no choice as to which columns in a table were displayed, the left to right display order of those columns, the width of a column and how the rows in a table were sorted. As of NASIS 4.1, a user could modify all of these options. NASIS provides a special editor for the creation of Edit Setups.

A user can assign a name to an Edit Setup, but that name must be unique within the context of the NASIS site in which that user is working. A user with write access can create any number of Edit Setups, and a user can specify a particular Edit Setup as their default Edit Setup.

A number of exceptions in the course of a NASIS session can cause a user's current Edit Setup to revert to the NASIS default Edit Setup. For example, a user creates an Edit Setup that excludes a required field in a table, and then inserts a record into that table. When the user attempts to exit that table, an exception will occur because the required field was not entered. The system will change that user's current Edit Setup to the default Edit Setup (where all columns are visible), and place the cursor in the now visible required field.

NASIS 5.2 Edit Setup Owned Object



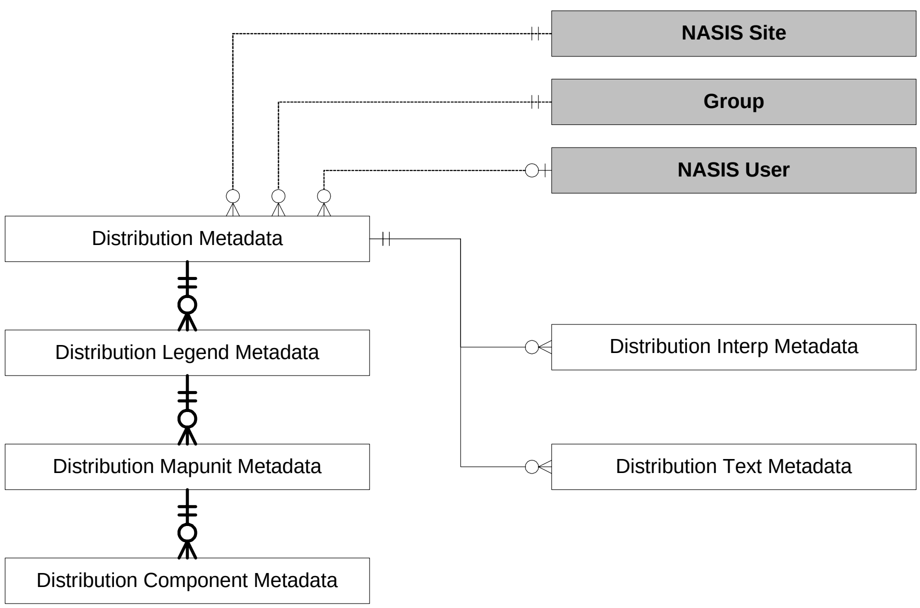
Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Distribution Metadata Owned Object

The Distribution Metadata owned object records the criteria used to generate a NASIS SSURGO export or Data Warehouse export, and includes the list of fuzzy logic interpretations included in that export, which types of narrative text were included in that export, as well as information about the legends, map units, data mapunits and components included in that export.

The Distribution Metadata owned object is populated exclusively by the NASIS SSURGO export process or Data Warehouse export process. A NASIS user can read this object, but they cannot modify its data and cannot manually add new instances. A NASIS user with the appropriate permissions can delete an instance of this owned object.

NASIS 5.2 Distribution Metadata Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

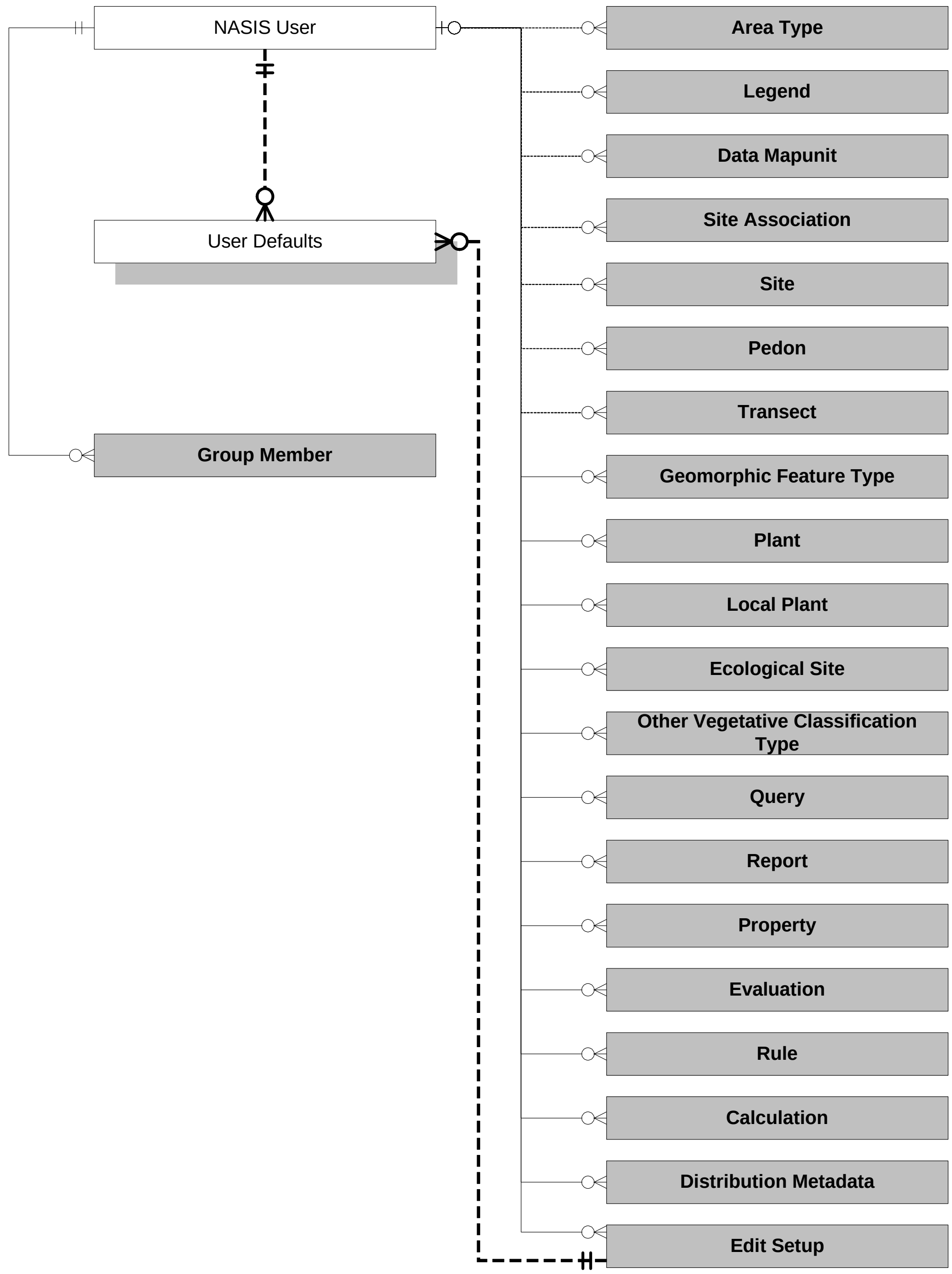
NASIS User Owned Object

Prior to NASIS 5.0, a NASIS user only existed within the context of a NASIS Site. As of NASIS 5.0, a NASIS user is now defined independent of any particular NASIS Site. One reason that this change was possible was because in version 5.0, NASIS became a centralized database, as opposed to a distributed database.

Prior to NASIS 5.0, if a NASIS user, say someone in a state office, needed to access data from more than one NASIS site, that user had to be added as a user for each NASIS site they needed to access. As of NASIS 5.0, a particular NASIS user can be a member of any number of NASIS sites, without having to duplicate that user for each of those NASIS sites.

Because the NASIS User owned object is independent of any particular NASIS site, its security is handled differently from that of all other NASIS owned objects. A new NASIS user can no longer be added using the NASIS software. New NASIS users must be added into the NASIS User table by a NASIS database administrator. Once a new NASIS user had been added by a NASIS database administrator, only that NASIS user can then update their user record.

NASIS 5.2 NASIS User Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

NASIS Site Owned Object

When NASIS was a distributed database, prior to NASIS 5.0, the concept of a NASIS Site was necessary. Each NASIS database represented a unique NASIS Site. All unique constraints were in the context of a NASIS Site. This allowed data from multiple NASIS Sites to be merged into a single database. The original NASIS security scheme was also centered around the concept of a NASIS Site. All data in NASIS was owned by one and only one NASIS site.

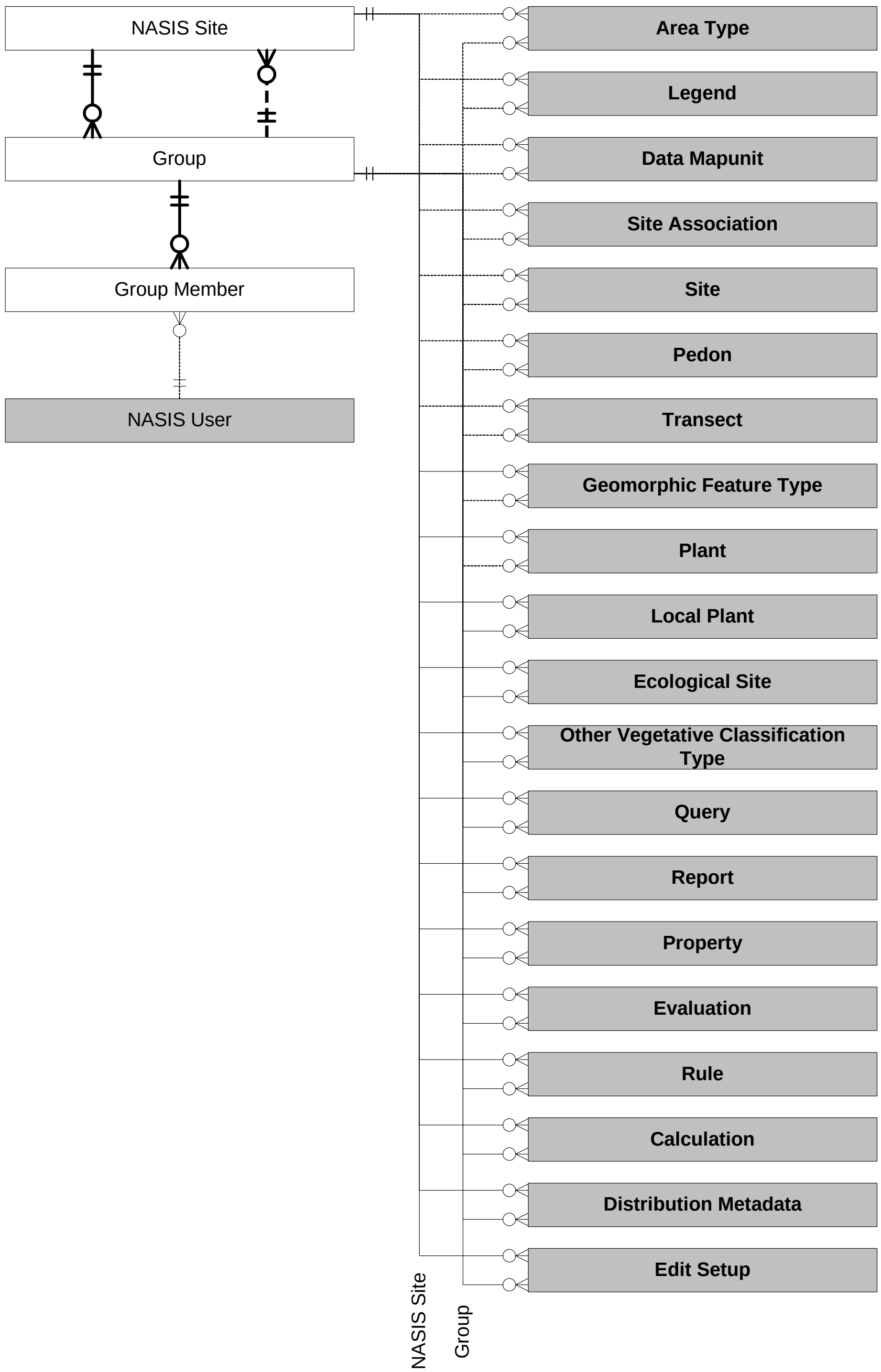
As of NASIS 5.0, the NASIS database became a centralized database. This made it possible to now establish unique constraints that were not constrained within the context of a NASIS site. But because so much existing data was only unique within the context of a NASIS Site, and because we still needed some organizational unit concept for controlling access to data, the NASIS Site concept was retained. A NASIS Site is now a logical construct, as opposed to a physical database location.

Do not confuse the NASIS Site owned object with the Site owned object. The Site owned object represents a point on the ground, somewhere on planet Earth. The NASIS Site owned object represents an NRCS organizational unit, most often an MLRA office.

Every instance of every NASIS owned object, with the exception of the NASIS User owned object and the NASIS Site owned object itself, must be associated with one and only one NASIS site, and one and only one group that is a member of that NASIS site.

The NASIS Site owned object is used to implement security and access control in a NASIS database. In order to be able to modify an instance of an owned object in NASIS, a person must be a valid NASIS user who is a member of the group that currently owns that instance.

NASIS 5.2 NASIS Site Owned Object



Clear Entity - Member of Owned Object Being Diagrammed
Clear Shadowed Entity - Hidden Member of Owned Object Being Diagrammed
Shaded Entity - Member of Owned Object Other Than One Being Diagrammed
Shaded Shadowed Entity - Hidden Member of Owned Object Other Than One Being Diagrammed
Normal Relationship - Table Up / Down Relationship
Heavy Relationship - Default Down Table Relationship
Dashed Relationship - Load / Find Related Relationship
Heavy Dashed Relationship - Non-Traversable Relationship

Major Load Related/Find Related Relationships

In NASIS, a relationship between two tables falls into one of two NASIS defined categories. An “edit” relationship exists between two tables if they are members of the same owned object, and, if, in the NASIS editor, one navigates between those two tables using the “down table” and “up table” functions. Any relationship between two tables that is not an “edit” relationship, we define as a “link” relationship. The two tables involved in a link relationship do not have to be members of the same owned object, but they may be.

In most, but not all, cases, a “link” relationship is a lookup from one table into the related table. For example, in the correlation table, one can open a choice list in order to establish a relationship with some data mapunit. Not all “link” relationships in a NASIS database can be navigated. In some cases, one or both of the tables involved in a “link” relationship may be hidden. In other cases, there may be more than one “link” relationship between the same two tables. At this point in time, NASIS is only capable of navigating one of the possible multiple “link” relationships between a particular pair of tables.

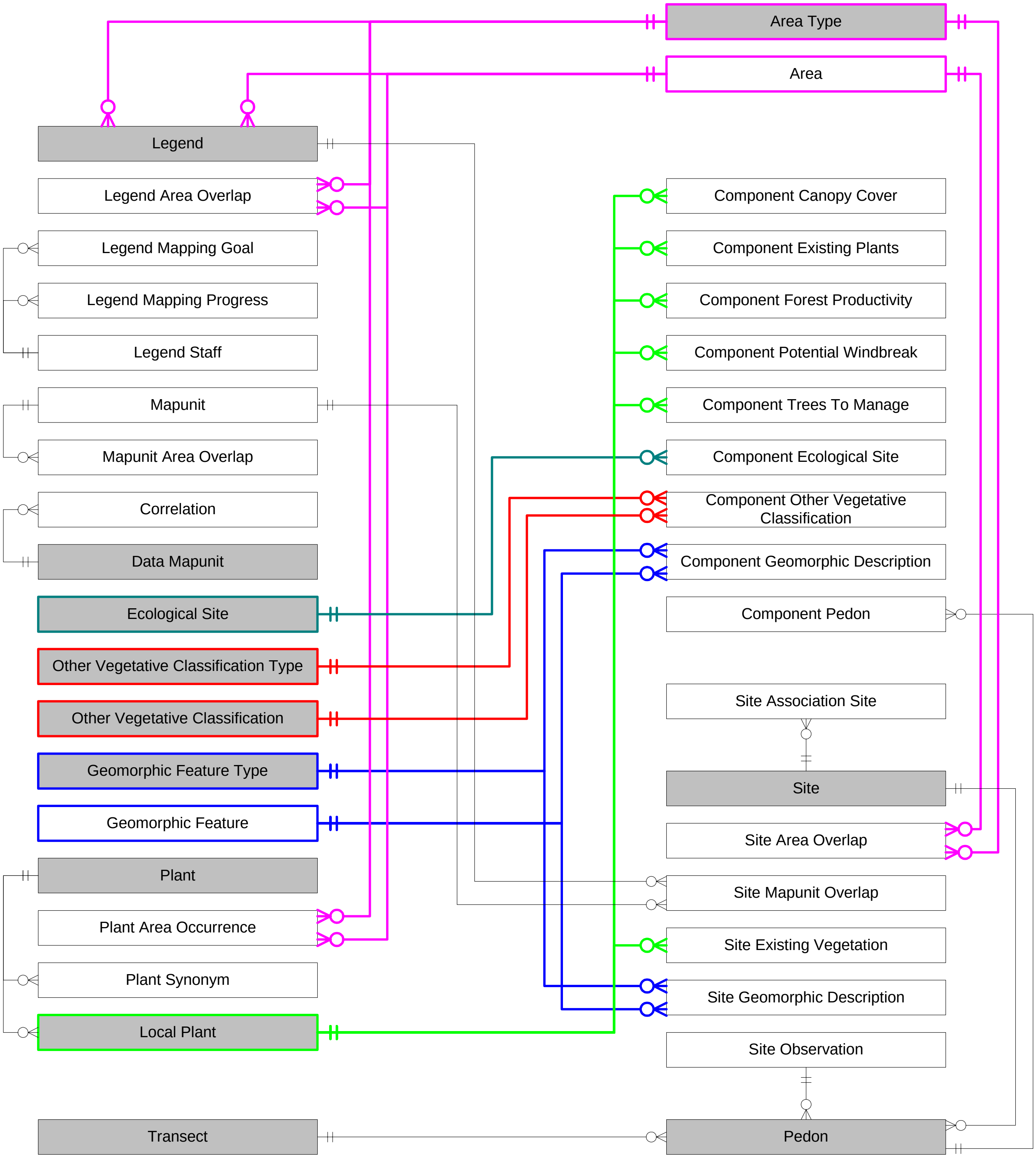
Navigable “link” relationships cannot be traversed using the “down table” and “up table” functions. Navigable “link” relationships are traversed using the “load related” and “find related” functions in NASIS.

The “load related” function loads all records into a target table that are related to a selected record or selected set of records in the source table. The “load related” function loads records into the NASIS editor, but does not change your current editor view by moving from the source table to the target table into which records may or may not have just been loaded.

The “find related” function attempts to find the first related record in a target table this is related to a selected record or selected set of records in the source table. In order to find a related record, that record must have already been loaded into the NASIS editor. Unlike “load related”, “find related” always changes your current editor view by moving from the source table to the target table, regardless of whether or not any related record was found.

Most of the “load related”/”find related” relationship in NASIS are shown on the following diagram. What is not shown are the relationships between every owned object root table and the NASIS Site, Group and NASIS User tables. Showing all of these relationships would have cluttered up this diagram beyond all usability. Keep in mind though that these navigable relationships do exist and are a handy way of loading all instances of a particular owned object that are owned by a particular NASIS site or group, or were last modified by a particular NASIS user.

NASIS 5.2 Major Load Related/Find Related Relationships



Clear Entity - Non-Root Table
Shaded Entity - Owned Object Root Table

A Primer on Bachman Data Model Diagram Symbology

There are a number of standards used for data model diagrams. The standard used for the diagrams in this document is referred to as Bachman notation. There are only two different types of symbols used in these diagrams, an entity symbol and a relationship symbol.

An entity is anything about which information can be stored; for example, a person, concept, physical object or event. In these diagrams, a rectangular box represents an entity. Entities typically correspond to nouns. Typically, the entity symbols on conceptual model represent composite entities. The entities on a conceptual model often resolve into more than one table in a physical database. An entity symbol on a physical data model has a 1:1 correspondence with a physical database table.

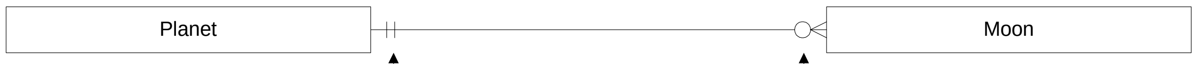
A relationship denotes some type of relationship between instances of one entity and, most often, instances of some other related entity. For example, the two entities “Company” and “Employee” are connected by a relationship that may be titled either “Works For” or “Employs”, depending on the direction in which the relationship is traversed. Relationships typically correspond to verbs. A relationship symbol is a line that typically connects one entity to another entity. It is also possible for an entity to have a recursive relationship with itself.

Both ends of a relationship line employ a special symbology. This symbology denotes two things, the cardinality of the relationship, and whether or not the relationship is optional or mandatory. The cardinality of a relationship indicates how many instances of one entity may be related to one instance of the other entity involved in the relationship. Cardinality is expressed on both ends of a relationship line. The concept of mandatory or optional indicates whether or not an instance of one entity can exist without having a relationship to at least one instance of the related entity. Again, this concept is denoted on both ends of a relationship line.

Many of the conventions used in these diagrams have nothing to do with Bachman notation. The Bachman notation has no concept of shaded or shadowed entities. The Bachman notation has no concept of different weights of relationship lines and no concept of dashed relationship lines. These conventions were added to illustrate specific aspects of how things are organized and managed in a NASIS database.

The best way to illustrate the concepts described above is with examples of entities and relationships that are familiar to everyone. The trick to stating a relationship rule is to always speak in terms of how a single instance of one entity is related to however many instances of the related entity. This phrasing should be employed in stating both the relationship from entity A to entity B, and the relationship from entity B to entity A.

Bachman Notation Examples



A moon must be a moon of one and only one planet. The relationship from moon to planet is mandatory, i.e. an instance of moon cannot exist without a corresponding instance of planet.

A planet may have zero to many moons. The relationship from planet to moon is optional, i.e. an instance of planet may exist with no corresponding instance of moon.

At any given moment in time, a kidney may reside in zero (on the way to be transplanted) or one person (Siamese twins not withstanding). The relationship from kidney to person, is optional, i.e. an instance of kidney may exist without a corresponding instace of person, at least for some period of time.

A person must have one or more kidneys. The relationship between person and kidney is mandatory, i.e. an instance of person cannot exist without a corresponding instance of kidney.

A geographic region may have zero to many native plant species. The relationship from geographic region to native plant species is optional, i.e. a instance of a geographic region may exist without any corresponding instance of native plant species.

A native plant species must be native to one or more geographic regions. The relationship from native plant species to geographic region is mandatory, i.e. an instance of a native plant speices cannot exist without an instance of a geographic region for which that plant species is native.

Note that a relationship rule is stated in a particular direction. In stating the relationship from left to right, the manner in which an instance of the left entity is related to instances of the right entity is denoted by the relationship symbol closest to the entity on the right. Of course the reverse is true when stating the relationship rule from right to left.